WEST Search History

| Hide Items | Restore | Clear | Cancel |

DATE: Monday, June 28, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L9	19980513	8
☐	L8	19980513	3
☐	L7	L6 and (remote adj4 (processing or computing))	9
☐	L6	((task or proceses or processing) adj4 agent) and ((co-ordinating or coordinating) adj4 agent)	42
☐	L5	19980513	2
☐	L4	(mobile adj3 agent) and (remote adj4 computing)	10
☐	L3	19980513	0
☐	L2	(mobile adj3 agent)and (task adj4 agent) and ((co-ordinating or coordinating) adj4 agent)	10
☐	L1	(mobile adj3 agent) same (remote adj4 computing)	1

END OF SEARCH HISTORY

☐ ▓▓▓ Generate Collection ▓▓▓

L5: Entry 1 of 2   File: USPT   Nov 14, 2000

DOCUMENT-IDENTIFIER: US 6148327 A
TITLE: Mobile agent docking arrangement for enhancing agent capabilities

Abstract Text (1):
Intelligent mobile agents are transmitted to computers at remote sites over a
network having a relatively limited bandwidth. In order to provide the mobile
agents with extended capabilities, yet to allow them to access information in near-
real-time, the intelligent mobile agents dock with docking systems at the network
nodes. The docking systems respond to the presence of the mobile agents, and
provide extended capabilities to the agent. The extended capabilities, in turn, are
made available to the various network nodes at which the agents may dock by loading
from portable data storage media, or preferably by occasional loading from the
network at off-peak times.

Application Filing Date (1):
19961105

Brief Summary Text (2):
This invention relates to distributed networks of computers linking computing and
data resources, and more particularly to such networks which utilize intelligent
mobile agents for performing complex tasks in the presence of limited network
bandwidth.

Brief Summary Text (4):
Distributed communication systems or networks are now widely used. Such
communications systems may be widely distributed information sources such as local-
or wide-area networks, corporate or military communication systems, or the world-
wide computerized interactive communication system known as the internet. Such
communication systems can be used to access remote information or computing
resources. When using such systems, it is necessary to know the location or address
of the information being sought or of the computing resource to be used, or the
user must carry out a search procedure to locate the information. In the case of
very large distributed systems which are not indexed, the search is so extensive or
time-consuming that it may be impractical, and may therefore not be attempted.

Brief Summary Text (5):
There has been interest in the use of intelligent mobile agents for overcoming some
of the problems associated with searches of distributed, possibly unindexed, data
bases or computing resources. An intelligent mobile agent is a computer program
which can independently or semi-independently perform tasks which the operator
could not perform on his own because of the time or effort required for the task.
For example, an intelligent mobile agent might be used to discover information
autonomously, or under its own control, because the operator or user of the
distributed communication system might not be aware of the existence of a source of
useful information, or of its address if its existence were known. The mobile
intelligent agent is a computer-type program characterized by the ability to move
over the communication system from one computer to another while the program is
"running"; in fact, it is necessary for it to do so in order to perform its tasks.
This does not, of course, mean that the bit pattern of the program changes while it
is in transit from one computer to the other, but rather means that, in performing

its tasks or "running," the program moves its location from one computer to another. Intelligent mobile agents or "softbots" are described in the following references:

Brief Summary Text (10):
5) "Telescript Technology: Mobile Agents", James E. White, General Magic, Inc., American Association for Artificial Intelligence.

Brief Summary Text (11):
One problem which has been found with the use of such mobile agents for seeking information on a distributed computing or knowledge base is that the capability of the agent is more or less related to the size or complexity of the program which embodies the intelligent mobile agent, so that a highly capable mobile agent, just like a highly capable computer program of any sort, tends to be very large in terms of the number of bytes which it contains. The length of time required to transmit the intelligent mobile agent from one computer to another over the communication system depends upon the bandwidth of the system in bits or bytes per second. It may not be practical to use a highly capable intelligent mobile agent, because the large size of such an agent requires excessive computer-to-computer transmission time. If the bandwidth of the distributed communication system can be controlled, then its bandwidth can simply be expanded in order to accommodate the desired mobile agents at the desired speed. Ordinarily, the communication system or data base is a given, and its bandwidth cannot be controlled. It should be noted here that limitations in the bandwidth of a part of a computer which interacts with the communication system will have the same effect, as to that computer, as a limitation of the bandwidth of the communication system itself; it is therefore desirable to use the highest possible speed modem at each computer site. Under bandwidth-limited conditions, the use of a highly capable intelligent mobile agent may result in excessive transmission time delays, and these directly impact the time required for the intelligent mobile agent to perform a search. The amount of time which is considered to be excessive may depend upon the information sought and its importance. If search time is irrelevant, a highly capable intelligent mobile agent may be used on a low-bandwidth distributed computing or network. Ordinarily, however, large transmission times are undesirable because of the resulting delays in receiving the search results, especially in those cases in which real-time information is sought. Thus, if a very complex search is required to find the desired information, a highly capable intelligent mobile agent capable of performing the search may be too large to be practical under given circumstances. The inability to quickly transmit highly capable intelligent mobile agents prevents full utilization of the resources of the communication system.

Brief Summary Text (12):
Improved methods are desired for transmitting highly capable intelligent mobile agents over communication systems interlinking distributed computing systems.

Brief Summary Text (14):
A distributed communication network includes a plurality of computers, and communication paths for coupling the computers together for communications therebetween. The network also includes standardized computer network software associated with each of the computers, for providing basic network communications between or among the computers; this may, in one embodiment of the invention, be software suitable for use with a world-wide communication system commonly known as "internet." The network according to the invention also includes a mobile agent generating program located at one or more of the computers of the network, for generating, and for transmitting over the communication network to at least one other computer of the network, agents for performing a desired function. The dimensions of a mobile agent are related to its capability, whereby highly capable agents may be so large as not to be usable, because of network bandwidth limitations. A mobile agent docking arrangement is located at each computer of the network, which capability expander is capable of coacting with the mobile agents,

for rendering the agents active at that computer. According to an aspect of the invention, a <u>mobile agent</u> capability expander is located at, or associated with, at least one of the computers of the network which is capable of coacting with <u>mobile agents,</u> for extending at least one capability of a <u>mobile agent</u>. This enhances the capability of <u>mobile agents</u> used in the system without expanding limited network bandwidth. In a particular embodiment of the invention, the information embodying the capability enhancement or expander is transmitted over the communication network to the agent-receiving computer(s) at infrequent intervals, during which the communication network is otherwise lightly loaded, so that the capability expander is available at such later time at which a <u>mobile agent</u> arrives. In another embodiment of the invention, the capability enhancement may be uploaded to the agent-receiving computer(s) from a portable memory, such as a floppy disk.

<u>Drawing Description Text</u> (2):
FIG. 1 is a simplified block diagram of a communication system which can use intelligent <u>mobile agents</u> in accordance with an aspect of the invention;

<u>Drawing Description Text</u> (5):
FIG. 4 is a simplified flow chart illustrating a typical intelligent <u>mobile agent</u>;

<u>Detailed Description Text</u> (2):
In FIG. 1, a communication system designated generally as 10 includes a network 12 and four representative computers 14a, 14b, 14c, and 14d connected to the network. As mentioned above, each computer in a prior-art system is capable of sending intelligent <u>mobile agents</u> out to any of the other three computers. Such <u>mobile agents</u> can be generated at each of computers 14 by the use of a language such as agent TCL, available from the computer science division of Dartmouth College, Dartmouth, N.H., and can be transmitted, by way of the network 12, to the other computers after its generation. The bandwidth of a network such as network 14 varies from about 4800 baud or bits-per-second to as much as tens of megabaud, and may vary within the network, depending upon which network branch is considered.

<u>Detailed Description Text</u> (3):
FIG. 2 is a simplified block diagram of an arrangement according to the invention, which is similar to a portion of the arrangement of FIG. 1. In FIG. 2, a source of intelligent <u>mobile agents</u> is illustrated as 14a, which is the same designation as that of computer 14a of FIG. 1. Element 14a of FIG. 2 is designated as a "source", because in the most basic system, it needs only to transmit or launch a <u>mobile agent</u>. The active agent goes onto the network from element 14a of FIG. 2, and performs its tasks, moving from computer to computer as necessary. If the <u>mobile agent</u> is programmed to return to source 14a when its task is completed, then source 14a must also be capable of receiving the information conveyed by the <u>mobile agent</u>. While source/sink 14a of FIG. 2 need not be a computer, it will ordinarily be such.

<u>Detailed Description Text</u> (4):
Also in FIG. 2, network 12 is connected to a second "computer" block 14b, which may include a database 210 and an agent dock 212. Database 210 may be simply a memory which can be accessed by an intelligent <u>mobile agent,</u> in which case block 14b must further include a processor or "computer", because the intelligent <u>mobile agent</u> must reside, at least temporarily, at location 14b. Instead of a database such as 210, agent dock 212 could be coupled to any other data or computing resource which is to be acted upon by an intelligent <u>mobile agent</u>. As also illustrated in FIG. 2, database 210 is coupled to agent dock 212, and the network 12 is also connected to agent dock 212.

<u>Detailed Description Text</u> (5):
The intelligent <u>mobile agent</u> is a program. In an ordinary computer system, blocks 210 and 212 of location 14b of FIG. 2 may be hardware, or they may be programs (software) which are resident at location 14b. Agent dock 212 of FIG. 2 has the

ability to interface or interact with the intelligent mobile agent, and also
provides the intelligent mobile agent with capabilities which reside at location
14b. Such capabilities might include access to knowledge bases, additional
computational functions, and the like.

Detailed Description Text (6):
FIG. 3 is a simplified architecture diagram which represents the software in the
agent dock 212 of FIG. 2. In FIG. 3, the intelligent mobile agent 310 appears at
the bottom of block 212, adjacent to the path 214 by which it arrives at the dock.
When docked, mobile agent 310 interfaces with a docking software layer 312, which
recognizes the presence of the mobile agent, and which enables the mobile agent by
effectively "pushing its ON button", to cause the intelligent mobile agent to
recognize that it has arrived at a new location, and to enable it to execute its
program or instructions. Location 14b also includes a knowledge base illustrated as
314.

Detailed Description Text (7):
Following the turn-on or enabling of the then-resident intelligent mobile agent 310
of FIG. 3, the docking layer 312 acts as a service provider to the intelligent
mobile agent 310. For example, the intelligent mobile agent may indicate to the
docking layer that the mobile agent is a search agent which is looking for
particular data. Such data, in a military context, might be the presence, location,
or identity, or all of this information, relating to electromagnetic radiators
which might have been located by specialized equipment available at location 14b.
It may happen that the computer at location 14b has no specialized equipment for
generating the type of data which the intelligent mobile agent seeks, in which case
the docking layer 312 of FIG. 3 so advises the agent 310. In that event, the agent
may go elsewhere in the network (to another computer location 14a, 14b, 14c, 14d, .
. . ; of FIG. 1), or return to its source with a failure message. On the other
hand, if the docking layer 312 knows that such information is available in its
knowledge base 314, it advises the intelligent mobile agent 310 of that fact, and
of other relevant capabilities available. For example, in the context of an
electromagnetic emitter locator, location 14b might have a database 316 containing
raw data representing frequency and pulse characteristics of a received
electromagnetic signal, and might also have an expert system 318 which is capable
of evaluating the raw data, and which may have already evaluated the raw data, and
have identified an emitter as being, for example, a weather radar operating at a
particular physical location (which, in general, is not the same as a network
location). On the other hand, location 14b might have the raw data represented by
database 316, and the knowledge base required as the foundation for an expert
system for identifying the emitter, but might not include the expert system itself.

Detailed Description Text (8):
In general, expert systems include two distinct portions, a knowledge base and a
decision-making portion. The decision-making portion is typically much smaller than
the knowledge base. However, the knowledge base changes from time to time as more
information becomes available, and as conditions change. For example, in the
medical fields the knowledge about the symptoms and progress of newly discovered
diseases may not be in an older knowledge base, but, when such information is
included in an updated knowledge base, the same types of decision-making programs
can operate on the updated information in order to identify, and suggest treatment
for, such newly discovered diseases. Similarly, in the abovementioned military
application, the knowledge base 318d of FIG. 3 can be updated with the operating
frequency, pulse characteristics, and other spectral characteristics of new radar
and other emitter types as they become known, which the decision-making portion
318s of the expert system can then use to identify this new type of radar if the
raw data is indicative of its presence. Layer portion 320 represents other
capabilities which might be available to the intelligent mobile agent.

Detailed Description Text (9):
Thus, the docking layer 312 of FIG. 3 "tells" the then-resident <u>mobile agent</u> of
layer 310 about the existence of the raw data available in database 316, and of the
existence of either the knowledge base 318d of, or of the entire expert system 318,
and also about any other available capabilities 320 which are relevant to the task
which the intelligent <u>mobile agent</u> has identified.

Detailed Description Text (10):
Once the intelligent <u>mobile agent</u> 310 of FIG. 3 has received information from the
docking layer about the available capabilities, the intelligent <u>mobile agent</u>
decides, based upon its program, which of the locally available capabilities it
wishes to avail itself of. The docking layer facilitates obtaining the information
from the raw-data database or the expert system, or from a stand-alone knowledge
base, as required.

Detailed Description Text (11):
The information in expert system database 312 of FIG. 3 can be updated in any of a
number of ways. For example, the database 312 can be updated by occasional
transmissions over the network 12. While these transmissions may require
substantial bandwidth, they are performed infrequently, and may be performed at low
data rates, over extended intervals during low-traffic times. In the arrangement of
FIG. 3, the expert system and a current database therefor are available to the
intelligent <u>mobile agent,</u> without transmitting the entire expert system and its
database each time an inquiry is made by transmission of an intelligent <u>mobile
agent</u>. The knowledge base 318d may be updated by a memory upload provided by a
maintenance worker from a memory disk, or from any other physical media.

Detailed Description Text (12):
FIG. 4 is a simplified flow chart illustrating the operation of a typical <u>mobile
agent</u> 310 in accordance with the invention, and FIG. 5 is a simplified flow chart
illustrating the operation of the docking system 312. The sequence of events begins
with the arrival of a <u>mobile agent</u> at a network node (network site) or an agent
dock, as suggested by block 410 of FIG. 4. The sequence representation then flows
to a logic node 5-1, which represents a transfer of the sequence to a corresponding
logic node 5-1 of FIG. 5. Logic block 510 represents a start point for the docking
program, which starts up the logic, and then goes into a "wait" state, waiting for
arrival of an agent at the node. From logic node 5-1 of FIG. 5, the logic flows to
a block 512, which represents recognition by the logic flow of the docking system
312 that a <u>mobile agent</u> has arrived at the network node. The recognition
represented by block 512 may be accompanied by a verification of a security code,
following which the docking system logic flows to a decision block illustrated as
514. Decision block 514 decides whether the currently arrived agent has clearance
to coact with this network node. If the <u>mobile agent</u> does not meet the security
requirement, the logic leaves decision block 514 by the NO output, and flows to a
block 518, which represents rejection of the <u>mobile agent</u>. The logic of the agent
docking system then returns by a logic path 419 to block 512, whereupon the docking
system continues to monitor for the appearance of an active agent at its network
node.

Detailed Description Text (13):
If a <u>mobile agent</u> arrives at the network node and meets the established security
criteria, the logic leaves decision block 514 of FIG. 5 by way of the YES output,
and flows to a block 516, which represents a command for the starting of the <u>mobile
agent</u>. From block 516, the logic flows by way of a logic node 4-2 to block 412 of
FIG. 4. Block 412 of FIG. 4 represents the starting of the <u>mobile agent</u> in response
to the command issued by logic block 516. From logic block 412, the logic flows to
a decision block 414, which decides whether the <u>mobile agent</u> requires capability
extension. This decision may be as simple as the examination of an "extension
required" flag associated with the <u>mobile agent</u>. If no capability extension is
required, the logic flows from decision block 414 by way of its NO output, and

arrives at a block 416. Block 416 represents performance of the task assigned to the mobile agent by its internal programming. Once the task is completed, the logic flows from block 416 to a block 418. Since the mobile agent has completed its task, it must now decide whether it must return to the originating source with its data, or to seek more data by going on to a known further address, or search randomly by going to an address which has not yet been visited. Block 418 represents the decision as to where the mobile agent is to go next. From block 418, the logic flows to a block 420, which represents an instruction to the docking system to command the transmission of the mobile agent to the selected address. From logic block 420 of FIG. 4, the logic then flows by way of a logic node 5-3 to logic block 520 of FIG. 5, which represents acceptance of the direction to move the mobile agent to a defined network node. Block 522 represents the sending of the mobile agent to its next destination, and the logic then flows by way of a path 523 back to block 512.

Detailed Description Text (14):
In the event that the mobile agent requires extension of its capabilities, the logic flow is as described above until the logic arrives at decision block 414 of FIG. 4. In the case in which the mobile agent requires capability extension, the logic leaves decision block 414 by the YES output, and flows to a block 422, which represents making a request for a particular extension of capabilities from the docking system. It must be recognized that the network may have many nodes having different capabilities, and may have various kinds of mobile agents perambulating therethrough, seeking various different types of capabilities, not all of which are available at each network node. From block 422 of FIG. 4, the logic flows by way of a logic node 5-2 to a block 524 of FIG. 5. Block 524 represents the reception of a request for a particular type of extension capability. From block 524, the logic of the docking system flows to a decision block 526, which looks through its index of available capabilities to see if the requested capability is available. If the capability is not available at this particular network node, the logic leaves decision block 526 by the NO output, and flows to a block 528, which represents the sending of a "not available" signal to the mobile agent. From block 528 of FIG. 5, the control flows by way of a logic node 4-3 to decision block 424 of FIG. 4. Since the capability sought by the mobile agent is not available at this particular network node, the logic of the mobile agent leaves decision block 424 by the NO output, and flows to block 418. As described above, block 418 represents the determination of the location of the next network node to be visited by the mobile agent; from block 418, the logic passes through the states represented by logic blocks or nodes 420, 5-3, 520, and 522, back to block 512, which represents a "waiting for a mobile agent" state of the docking system.

Detailed Description Text (15):
As so far described, the cases which have been discussed in relation to FIGS. 4 and 5 are those in which (a) the mobile agent was not authorized to act, and was rejected; (b) was authorized to act, but did not require capability extension; (c) was authorized to act, and required capability extension, but the particular capability extension was not available at the network node or site. The next possibility is that the mobile agent arriving at the network node is authorized, requires capability, and the capability is available. In this last case, the logic will arrive at decision block 526 of FIG. 5 as described above, but, since the requested capability is available, the logic will leave by the YES output, rather than NO. From the YES output of decision block 526, the logic flows to a block 530, which represents the passing of a capability pointer to the requesting mobile agent. Logic node 4-3 returns the sequence flow to decision block 424 of FIG. 4, whereupon the logic leaves decision block 424 by the YES output, and flows to block 416, which as mentioned above represents the performance of the task, which presumably also means availing itself of the extended capabilities at the network node.

Detailed Description Text (17):

FIG. 6 is a simplified block diagram illustrating three different arrangements for
loading extended capability information into a computer at a remote location on the
network so that it may be accessed by the docking system of that computer in
response to the docking of a mobile agent. In FIG. 6, representative node 14d
includes a medium reader, such as a floppy-disk reader 610, which is coupled to a
capabilities repository, corresponding to the database, expert system, or the like
of FIG. 3. Updates of the capability can be manually transported to the node on a
data medium, illustrated as a floppy disk 612, and uploaded by means of the reader
610. Another way to update the knowledge base is by a mobile agent, illustrated as
Loader Agent 614, which is especially adapted for uploading or deliver of such
information from a remote site during periods of low traffic on the network.
Instead of using a mobile agent to control the updating, control could be
accomplished manually by an operator at a remote site, with the delivery of the
update information being performed over the network. It will be clear that updates
over the network must be accomplished in the presence of sufficient security to
guarantee that unwanted intrusion does not occur. The physical transmission by way
of a data medium may also require security measures to assure that the knowledge
base is not updated by someone having physical access but without authority to
update.

Detailed Description Text (18):
Thus, a distributed communication network (10) according to the invention includes
a plurality of computers (12), and communication paths (14) for coupling the
computers (12) together for communications therebetween. The network also includes
standardized computer network software, such as TCP/IP, IP, NFS, SLIP, and the
like, associated with each of the computers (12), for providing basic network
communications between or among the computers; this may, in one embodiment of the
invention, be software, such as TCP/IP, suitable for use with a world-wide
communication system commonly known as "internet." The network according to the
invention also includes a mobile agent generating program, such as Agent TCL,
located at one or more of the computers (12) of the network (10), for generating,
and transmitting over the communication means (14) to at least one other computer
of the network, an agent for performing a desired function. The dimensions of a
mobile agent are related to its capability, whereby highly capable agents may be so
large as not to be usable because of network bandwidth limitations. A mobile agent
docking arrangement (312) is located at each computer (14a, 14b, 14c, 14d) of the
network (10) which is capable of coacting with the mobile agent, for rendering the
mobile agent active at that computer. According to an aspect of the invention, a
mobile agent capability expander (312, 318) is located at at least one of the
computers (14b) of the network (10) which is capable of coacting with mobile
agents, for extending at least one capability of a mobile agent. This enhances the
capability of mobile agents used in the system without expanding limited network
bandwidth.

Detailed Description Text (19):
Other embodiments of the invention will be apparent to those skilled in the art.
For example, any of various means might be employed to realize network
communications, generation of mobile agents, and installation and update of agent-
extension capabilities at the nodes of the network. Networking communications among
nodes might be implemented by means other than hard-wired TCP/IP links. Such
variants might include wireless-based or infrared systems, asynchronous
transmission mode (ATM) systems, or serialized approaches such as SLIP and PPP.
Also, other languages and mechanisms for specifying and generating mobile agents
may be used. For example, Telescript, originating from General Magic, Inc., or
Java, originating from Sun Microsystems, could be used. Agent generation may also
be accomplished by a generating program built on, for example, the Unix Operating
System Remote Procedure Call.

Other Reference Publication (1):
"Telescript Technology: Mobile Agents" by James E. White, General Magic, Inc.

published by Amer. Assoc. for Artificial Intelligence,1996.

CLAIMS:

1. A distributed communication network, comprising:

a plurality of computers;

communication means for coupling said computers together for communications therebetween;

standardized computer network software associated with each of said computers, for providing basic network communications between said computers by way of said communication means;

mobile agent generating means located at at least one of said computers of said network, for generating mobile agents for performing a desired function, and for transmitting said mobile agents over said communication means, the dimensions of each of said mobile agent being generally related to its capability, whereby highly capable mobile agents may not be usable because of network bandwidth limitations;

mobile agent docking means located at each computer of said network which is capable of coacting with said mobile agents, for rendering said agents active at said coacting computer; and

mobile agent capability expanding means located at at least one of said coacting computers of said network, for extending at least one capability of a mobile agent received at said one of said coacting computers of said network, whereby the capability of said mobile agent may be enhanced.

2. A communication network according to claim 1, wherein one of said communication mobile agent generating means and one of said mobile agent docking means are located together at at least some of said computers of said network.

3. A method for operating a communication system, comprising the steps of:

coupling together a plurality of computers, located at different sites, by way of a network defining a bandwidth, which bandwidth may differ at various locations throughout said network;

communicating among said computers by way of said network, using communication software at each said computer which adheres to a communication standard;

at at least one of said sites, generating mobile agents for performing a desired function, and transmitting said mobile agents onto said network to at least one other computer, whereby said bandwidth of said network may prevent a highly capable mobile agent from being transmitted and received in a suitably short length of time;

preloading into said at least one other computer a mobile agent docking means, which is capable of rendering said mobile agents active at said one of said sites;

preloading into said at least one other computer a mobile agent capability expanding means, for extending at least one capability of a mobile agent received at said one of said computers of said network, whereby the capability of said mobile agent may be enhanced.

First Hit · Fwd Refs

L5: Entry 2 of 2   File: USPT   Dec 28, 1999

DOCUMENT-IDENTIFIER: US 6009456 A
TITLE: Information exchange by intelligent mobile agents in a network

Abstract Text (1):

Intelligent mobile agents in a distributed network carry information tokens among
network nodes. Each token includes compressed information or data, and an
indication of the compression algorithm which can decompress the information. At
least some of the nodes have at least one decompression algorithm. A copy of a
token may be left at a node by an intelligent mobile agent moving through the
network. A later-arriving intelligent mobile agent can interrogate the node to
determine what information it has in the form of tokens, and can use the token
information, or take the token (or a copy) elsewhere.

Application Filing Date (1):
19970730

Brief Summary Text (2):
This invention relates to distributed networks of computers linking computing and
data resources, and more particularly to such networks which utilize cooperating
intelligent mobile agents for performing complex tasks in the presence of limited
network bandwidth.

Brief Summary Text (4):
Distributed communication systems or networks are now widely used. Such
communications systems may be widely distributed information sources such as local-
or wide-area networks, corporate or military communication systems, or the world-
wide computerized interactive communication system known as the internet. Such
communication systems can be used to access remote information or computing
resources. When using such systems, it is necessary to know the location or address
of the information being sought or of the computing resource to be used, or the
user must carry out a search procedure to locate the information. In the case of
very large distributed systems which are not indexed, the search is so extensive or
time-consuming that it may be impractical, and may therefore not be attempted.

Brief Summary Text (5):
There has been interest in the use of intelligent mobile agents for overcoming some
of the problems associated with searches of distributed, possibly unindexed, data
bases or computing resources. An intelligent mobile agent is a computer program
which can independently or semi-independently perform tasks which the operator
could not perform on his own because of the time or effort required for the task.
For example, an intelligent mobile agent might be used to discover information
autonomously, or under its own control, because the operator or user of the
distributed communication system might not be aware of the existence of a source of
useful information, or of its address if its existence were known. The mobile
intelligent agent is a computer-type program characterized by the ability to move
over the communication system from one computer to another while the program is
"running"; in fact, it is necessary for it to do so in order to perform its tasks.
This does not, of course, mean that the bit pattern of the program changes while it
is in transit from one computer to the other, but rather means that, in performing
its tasks or "running," the program moves its location from one computer to

another. Intelligent <u>mobile agents</u> or "softbots" are described in the following references:

Brief Summary Text (10):
5) "Telescript Technology<u>: Mobile Agents</u>", James E. White, General Magic, Inc., American Association for Artificial Intelligence.

Brief Summary Text (11):
One problem which has been found with the use of such <u>mobile agents</u> for seeking information on a distributed computing or knowledge base is that the capability of the agent is more or less related to the size or complexity of the program which embodies the intelligent <u>mobile agent,</u> so that a highly capable <u>mobile agent,</u> just like a highly capable computer program of any sort, tends to be very large in terms of the number of bytes which it contains. Thus, autonomous agents which maintain their state over time, and which are not totally "scripted" and therefore are knowledge-based or "learning" programs, as are the domains of knowledge about which they maintain an understanding, tend to be quite large. The length of time required to transmit the intelligent <u>mobile agent</u> from one computer to another (from one "site" or network "node" to another) over the communication system depends upon the bandwidth of the system in bits or bytes per second. It may not be practical to use a highly capable intelligent <u>mobile agent,</u> because the large size of such an agent requires excessive computer-to-computer transmission time. If the bandwidth of the distributed communication system can be controlled, then its bandwidth can simply be expanded in order to accommodate the desired <u>mobile agents</u> at the desired speed. Ordinarily, the communication system or data base is a given, and its bandwidth cannot be controlled. It should be noted here that limitations in the bandwidth of a part of a computer which interacts at a network node with the communication system will have the same effect, as to that computer, as a limitation of the bandwidth of the communication system itself; it is therefore desirable to use the highest possible speed modem at each computer site. Under bandwidth-limited conditions, the use of a highly capable intelligent <u>mobile agent</u> may result in excessive transmission time delays, and these directly impact the time required for the intelligent <u>mobile agent</u> to perform a search. The amount of time which is considered to be excessive may depend upon the information sought and its importance. If search time is irrelevant, a highly capable intelligent <u>mobile agent</u> may be used on a low-bandwidth distributed computing system or network. Ordinarily, however, large transmission times are undesirable because of the resulting delays in receiving the search results, especially in those cases in which real-time information is sought. Thus, if a very complex search is required to find the desired information, a highly capable intelligent <u>mobile agent</u> capable of performing the search may be too large to be practical under given circumstances. The inability to quickly transmit highly capable intelligent <u>mobile agents</u> tends to prevent full utilization of the resources of the communication system.

Brief Summary Text (12):
U.S. patent application Ser. No. 08/741,759, filed Nov. 5, 1996 in the name of Whitebread et al., describes a scheme for improving the resource utilization in limited bandwidth communication systems. As described generally therein, a distributed communication network includes a plurality of computers, and communication paths for coupling the computers together at nodes of the network for communications therebetween. The network also includes standardized computer network software associated with each of the computers, for providing basic network communications between or among the computers; this may be software suitable for use with a world-wide communication system commonly known as "internet." The network according as described in the Whitebread et al. application also includes a <u>mobile agent</u> generating program located at one or more of the computers of the network, for generating, and for transmitting over the communication network to at least one other computer of the network, agents for performing a desired function. The dimensions (size) of a <u>mobile agent</u> are related to its capability, whereby highly capable agents may be so large as not to be usable, because of network

bandwidth limitations. A <u>mobile agent</u> docking arrangement is located at each computer or node of the network, which capability expander is capable of coacting with the <u>mobile agents,</u> for rendering the agents active at that computer. A <u>mobile agent</u> capability expander is located at, or associated with, at least one of the computers of the network which is capable of coacting with <u>mobile agents,</u> for extending at least one capability of a <u>mobile agent</u>. This enhances the capability of <u>mobile agents</u> used in the system without expanding limited network bandwidth. The information embodying the capability enhancement or expander may be transmitted over the communication network to the agent-receiving computer(s) at infrequent intervals, during which the communication network is otherwise lightly loaded, so that the capability expander is available at such later time at which a <u>mobile agent</u> arrives. The capability enhancement may instead be uploaded to the agent-receiving computer(s) from a portable memory, such as a floppy disk.

Brief Summary Text (13):
FIG. 1 illustrates a communication system designated generally as 10, which includes a network 12 and four representative network nodes with computers 14a, 14b, 14c, and 14d connected to the network, as described in the abovementioned Whitebread patent application. As mentioned above, each computer is capable of sending intelligent <u>mobile agents</u> out to any of the other three computers. Such <u>mobile agents</u> can be generated at each of computers 14 by the use of a language such as agent TCL, available from the computer science division of Dartmouth College, Dartmouth, N.H., and the <u>mobile agent</u> can be transmitted, by way of the network 12, to the other computers after its generation. The bandwidth of a network such as network 12 varies from about 4800 baud or bits-per-second to as much as tens of megabaud, and may vary within the network, depending upon which network branch is considered.

Brief Summary Text (14):
FIG. 2 is a simplified block diagram of an arrangement which is similar to a portion of the arrangement of FIG. 1. In FIG. 2, a source of intelligent <u>mobile agents</u> is illustrated as 14a, which is the same designation as that of computer 14a of FIG. 1. Element 14a of FIG. 2 is designated as a "source", because in the most basic system, it needs only to transmit or launch a <u>mobile agent</u>; it may, of course, have other capabilities. The active agent goes out onto the network from element 14a of FIG. 2, and performs its tasks, moving from computer to computer as necessary. If the <u>mobile agent</u> is programmed to return to source 14a when its task is completed, then source 14a must also be capable of receiving the information conveyed by the <u>mobile agent</u>. While source/sink 14a of FIG. 2 need not be a computer, it will ordinarily be such.

Brief Summary Text (15):
Also in FIG. 2, network 12 is connected to a second "computer" block 14b, which may include a database 210 and an agent dock 212. Database 210 may be simply a memory which can be accessed by an intelligent <u>mobile agent,</u> in which case block 14b must further include a processor or "computer", because the intelligent <u>mobile agent</u> must reside, at least temporarily, at location 14b. Instead of a database such as 210, agent dock 212 could be coupled to any other data or computing resource which is to be acted upon by an intelligent <u>mobile agent</u>. As also illustrated in FIG. 2, database 210 is coupled to agent dock 212, and the network 12 is also connected to agent dock 212.

Brief Summary Text (16):
The intelligent <u>mobile agent</u> is a program. In an ordinary computer system, blocks 210 and 212 of location 14b of FIG. 2 may be hardware, or they may be programs (software) which are resident at location 14b. Agent dock 212 of FIG. 2 has the ability to interface or interact with the intelligent <u>mobile agent,</u> and also provides the intelligent <u>mobile agent</u> with capabilities which reside at location 14b. Such capabilities might include access to knowledge bases, additional computational functions, and the like.
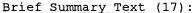
Brief Summary Text (17):
FIG. 3 is a simplified architecture diagram which represents the software in the
agent dock 212 of FIG. 2. In FIG. 3, one or more intelligent mobile agents 310a, .
. . , 310n appears at a layer 310 at the bottom of block 212, adjacent to the path
214 by which it (they) arrive at the dock. It should be recognized that at any
particular time, the agent dock may have no mobile intelligent agents docked
therewith, it may have only one, or it may have two or more. When docked, each
mobile agent 310a, . . . , 310n interfaces with a docking software layer 312, which
recognizes the presence of the mobile agent, and which enables the mobile agent by
effectively "pushing its ON button", to cause the intelligent mobile agent to
recognize that it has arrived at a new location, and to enable it to execute its
program or instructions. Location 14b also includes a knowledge base illustrated as
314.

Brief Summary Text (18):
Following the turn-on or enabling of the then-resident intelligent mobile agent
310a, . . . , 310n of FIG. 3, the docking layer 312 acts as a service provider to
the intelligent mobile agent 310a, . . . , 310n. For example, the (or one of the)
intelligent mobile agent(s) may indicate to the docking layer that the mobile agent
is a search agent which is looking for particular data. Such data, in a military
context, might be the presence, location, or identity, or all of this information,
relating to electromagnetic radiators which might have been located by specialized
equipment available at location 14b. It may happen that the computer at location
14b has no specialized equipment for generating the type of data which the
intelligent mobile agent seeks, in which case the docking layer 312 of FIG. 3 so
advises the agent 310a, . . . , 310n. In that event, the agent may go elsewhere in
the network (to another computer location 14a, 14b, 14c, 14d, . . . ; of FIG. 1),
or return to its source with a failure message. On the other hand, if the docking
layer 312 knows that such information is available in its knowledge base 314, it
advises the intelligent mobile agent 310a, . . . , 310n, of that fact, and of other
relevant capabilities available. For example, in the context of an electromagnetic
emitter locator, location 14b might have a database 316 containing raw data
representing frequency and pulse characteristics of a received electromagnetic
signal, and might also have an expert system 318 which is capable of evaluating the
raw data, and which may have already evaluated the raw data, and have identified an
emitter as being, for example, a weather radar operating at a particular physical
location (which, in general, is not the same as a network location). On the other
hand, location 14b might have the raw data represented by database 316, and the
knowledge base required as the foundation for an expert system for identifying the
emitter, but might not include the expert system itself.

Brief Summary Text (19):
In general, expert systems include two distinct portions, namely a knowledge base
and a decision-making portion. The decision-making portion is typically much
smaller than the knowledge base. However, the knowledge base changes from time to
time as more information becomes available, and as conditions change. For example,
in the medical fields the knowledge about the symptoms and progress of newly
discovered diseases and/or conditions may not be in an older knowledge base, but,
when such information is included in an updated knowledge base, the same types of
decision-making programs can operate on the updated information in order to
identify, and suggest treatment for, such newly discovered diseases and/or
conditions. Similarly, in the abovementioned military application, the knowledge
base 318d of FIG. 3 can be updated with the operating frequency, pulse
characteristics, and other spectral characteristics of new radar and other emitter
types as they become known, which the decision-making portion 318s of the expert
system can then use to identify this new type of radar if the raw data is
indicative of its presence. Layer portion 320 represents other capabilities which
might be available to the intelligent mobile agent.

Brief Summary Text (20):
Thus, the docking layer 312 of FIG. 3 "tells" the then-resident mobile agent or
agents of layer 310a, . . . , 310n, about the existence of the raw data available
in database 316, and of the existence of either the knowledge base 318d of, or of
the entire expert system 318, and also about any other available capabilities 320
which are relevant to the task which the intelligent mobile agent has identified.

Brief Summary Text (21):
Once the intelligent mobile agent 310a, . . . , 310n, of FIG. 3 has received
information from the docking layer about the available capabilities, the
intelligent mobile agent decides, based upon its program, which of the locally
available capabilities it wishes to avail itself of. The docking layer facilitates
obtaining the information from the raw-data database or the expert system, or from
a stand-alone knowledge base, as required.

Brief Summary Text (22):
The information in expert system database 318 of FIG. 3 can be updated in any of a
number of ways. For example, the database 318d can be updated by occasional
transmissions over the network 12. While these transmissions may require
substantial bandwidth, they are performed infrequently, and may be performed at low
data rates, over extended intervals during low-traffic times. In the arrangement of
FIG. 3, the expert system and a current database therefor are available to the
intelligent mobile agent, without transmitting the entire expert system and its
database each time an inquiry is made by transmission of an intelligent mobile
agent. The knowledge base 318d may be updated by a memory upload provided by a
maintenance worker from a memory disk, or from any other physical media.

Brief Summary Text (23):
FIG. 4 is a simplified flow chart illustrating the operation of a typical mobile
agent 310a, . . . , 310n, as described in the abovementioned Whitebread patent
application, and FIG. 5 is a simplified flow chart illustrating the operation of
the docking system 312. The sequence of events begins with the arrival of a mobile
agent at a network node (network site) or an agent dock, as suggested by block 410
of FIG. 4. The sequence representation then flows to a logic node 5-1, which
represents a transfer of the sequence to a corresponding logic node 5-1 of FIG. 5.
Logic block 510 represents a start point for the docking program, which starts up
the logic, and then goes into a "wait" state, waiting for arrival of an agent at
the node. From logic node 5-1 of FIG. 5, the logic flows to a block 512, which
represents recognition by the logic flow of the docking system 312 that a mobile
agent has arrived at the network node. The recognition represented by block 512 may
be accompanied by a verification of a security code, following which the docking
system logic flows to a decision block illustrated as 514. Decision block 514
decides whether the currently arrived agent has clearance to coact with this
network node. If the mobile agent does not meet the security requirement, the logic
leaves decision block 514 by the NO output, and flows to a block 518, which
represents rejection of the mobile agent. The logic of the agent docking system
then returns by a logic path 419 to block 512, whereupon the docking system
continues to monitor for the appearance of an active agent at its network node.

Brief Summary Text (24):
If a mobile agent arrives at the network node and meets the established security
criteria, the logic leaves decision block 514 of FIG. 5 by way of the YES output,
and flows to a block 516, which represents a command for the starting of the mobile
agent. From block 516, the logic flows by way of a logic node 4-2 to block 412 of
FIG. 4. Block 412 of FIG. 4 represents the starting of the mobile agent in response
to the command issued by logic block 516. From logic block 412, the logic flows to
a decision block 414, which decides whether the mobile agent requires capability
extension. This decision may be as simple as the examination of an "extension
required" flag associated with the mobile agent. If no capability extension is
required, the logic flows from decision block 414 by way of its NO output, and

arrives at a block 416. Block 416 represents performance of the task assigned to the <u>mobile agent</u> by its internal programming. Once the task is completed, the logic flows from block 416 to a block 418. Since the <u>mobile agent</u> has completed its task, it must now decide whether it must return to the originating source with its data, or to seek more data by going on to a known further address, or search randomly by going to an address which has not yet been visited. Block 418 represents the decision as to where the <u>mobile agent</u> is to go next. From block 418, the logic flows to a block 420, which represents an instruction to the docking system to command the transmission of the <u>mobile agent</u> to the selected address. From logic block 420 of FIG. 4, the logic then flows by way of a logic node 5-3 to logic block 520 of FIG. 5, which represents acceptance of the direction to move the <u>mobile agent</u> to a defined network node. Block 522 represents the sending of the <u>mobile agent</u> to its next destination, and the logic then flows by way of a path 523 back to block 512.

Brief Summary Text (25):
In the event that the <u>mobile agent</u> requires extension of its capabilities, the logic flow is as described above until the logic arrives at decision block 414 of FIG. 4. In the case in which the <u>mobile agent</u> requires capability extension, the logic leaves decision block 414 by the YES output, and flows to a block 422, which represents making a request for a particular extension of capabilities from the docking system. It must be recognized that the network may have many nodes having different capabilities, and may have various kinds of <u>mobile agents</u> perambulating therethrough, seeking various different types of capabilities, not all of which are available at each network node. From block 422 of FIG. 4, the logic flows by way of a logic node 5-2 to a block 524 of FIG. 5. Block 524 represents the reception of a request for a particular type of extension capability. From block 524, the logic of the docking system flows to a decision block 526, which looks through its index of available capabilities to see if the requested capability is available. If the capability is not available at this particular network node, the logic leaves decision block 526 by the NO output, and flows to a block 528, which represents the sending of a "not available" signal to the <u>mobile agent</u>. From block 528 of FIG. 5, the control flows by way of a logic node 4-3 to decision block 424 of FIG. 4. Since the capability sought by the <u>mobile agent</u> is not available at this particular network node, the logic of the <u>mobile agent</u> leaves decision block 424 by the NO output, and flows to block 418. As described above, block 418 represents the determination of the location of the next network node to be visited by the <u>mobile agent</u>; from block 418, the logic passes through the states represented by logic blocks or nodes 420, 5-3, 520, and 522, back to block 512, which represents a "waiting for a <u>mobile agent</u>" state of the docking system.

Brief Summary Text (26):
As so far described, the cases which have been discussed in relation to FIGS. 4 and 5 are those in which (a) the <u>mobile agent</u> was not authorized to act, and was rejected; (b) was authorized to act, but did not require capability extension; (c) was authorized to act, and required capability extension, but the particular capability extension was not available at the network node or site. The next possibility is that the <u>mobile agent</u> arriving at the network node is authorized, requires capability, and the capability is available. In this last case, the logic will arrive at decision block 526 of FIG. 5 as described above, but, since the requested capability is available, the logic will leave by the YES output, rather than NO. From the YES output of decision block 526, the logic flows to a block 530, which represents the passing of a capability pointer to the requesting <u>mobile agent</u>. Logic node 4-3 returns the sequence flow to decision block 424 of FIG. 4, whereupon the logic leaves decision block 424 by the YES output, and flows to block 416, which as mentioned above represents the performance of the task, which presumably also means availing itself of the extended capabilities at the network node.

Brief Summary Text (28):

FIG. 6 is a simplified block diagram illustrating three different arrangements for loading extended capability information into a computer at a remote location on the network so that it may be accessed by the docking system of that computer in response to the docking of a mobile agent. In FIG. 6, representative node 14d includes a medium reader, such as a floppy-disk reader 610, which is coupled to a capabilities repository, corresponding to the database, expert system, or the like of FIG. 3. Updates of the capability can be manually transported to the node on a data medium, illustrated as a floppy disk 612, and uploaded by means of the reader 610. Control could be accomplished manually by an operator at a remote site, with the delivery of the update information being performed over the network. It will be clear that updates over the network must be accomplished in the presence of sufficient security to guarantee that unwanted intrusion does not occur. The physical transmission by way of a data medium may also require security measures to assure that the knowledge base is not updated by someone having physical access but without authority to update. Another way to update the knowledge base is by a mobile agent, illustrated as Loader Agent 614, which is especially adapted for uploading or deliver of such information from a remote site during periods of low traffic on the network.

Brief Summary Text (29):
While it is very desirable to be able to express highly capable mobile agents at remote nodes, it is also advantageous to be able to use the cooperation of multiple intelligent mobile agents to perform various tasks. Cooperation of multiple agents implies the transfer of information between the agents. The bandwidth limitation of the communication network prevents the mobile agents from carrying large amounts of information. Improved methods of exchange of information between intelligent mobile agents is desired.

Brief Summary Text (31):
A method according to the invention performs processing or information exchange on a distributed network of nodes by, or with, two or more intelligent mobile agents. Each of the nodes at which processing may be accomplished includes an intermediary module or program, corresponding to a docking module as described in the abovementioned Whitebread et al. patent application. The method includes various steps. Among the steps of the method is the step of selecting one compression algorithm from among the compression algorithms of a suite of compression algorithms, to thereby identify a selected compression algorithm. A compressed representation of a piece of information is generated by using the selected one of the compression algorithms. An information token is generated by attaching to the compressed piece of information a tag representing the identity of the selected compression algorithm used in the step of generating the compressed representation of the piece of information. The information token is attached to a first intelligent mobile agent. The first intelligent mobile agent is transmitted from a first one of the nodes in the system, together with its attached token, to at least a second node of the network. From a node in the network, which may be the first node or another node, a second intelligent mobile agent is transmitted to at least the second node of the network, whereby the first and second intelligent mobile agents reside at the second one of the nodes of the network, either simultaneously or sequentially. At the second node, the token is at least copied, or possibly transferred, from the first intelligent mobile agent to the intermediary module of the second node. In the intermediary module of the second node, the piece of information in the token is decompressed using the identity tag of the selected compression algorithm, to thereby produce a reconstituted piece of information. The reconstituted piece of information is copied or transferred to the second intelligent mobile agent. The second intelligent mobile agent then uses the reconstituted information to perform whatever task it is assigned to perform.

Brief Summary Text (32):
In a particular embodiment of the invention, the first mobile agent arrives at the second node at a time at which the second mobile agent is not yet at the node, and

the method further includes the steps of, for the first mobile agent, resident at
the second node, determining that the second mobile agent may eventually arrive at
the second node, and deciding to leave a copy of the token at the second node. In
response to the decision to leave a copy of the token, the first mobile agent
leaves a copy of the token with the intermediary module at the second node. In a
preferred embodiment of the invention, the intermediary module of the second node
leaves the token compressed until that later time at which arrival is identified,
at the second node, of an intelligent mobile agent which can make use of the piece
of information. When arrival of such an intelligent mobile agent is identified, the
piece of information is decompressed to produce the decompressed piece of
information, using the identity tag of the selected compression algorithm to select
the appropriate decompression algorithm. The reconstituted piece of information is
then copied, or possibly transferred, to the second intelligent mobile agent.

Drawing Description Text (2):
FIG. 1 is a simplified block diagram of a communication system which can use
intelligent mobile agents, as described in the abovementioned Whitebread et al.
application;

Drawing Description Text (5):
FIG. 4 is a simplified flow chart illustrating an intelligent mobile agent;

Drawing Description Text (8):
FIG. 7 is a simplified flow chart of the operation of a dock in accordance with an
aspect of the invention, for co-acting with intelligent mobile agents which may
carry tokens from one location to another; and

Drawing Description Text (9):
FIG. 8 is a simplified flow chart of the operation of an intelligent mobile agent
in accordance with an aspect of the invention, which may carry one or more tokens
from one location to another.

Detailed Description Text (3):
A user wishing to transmit an intelligent mobile agent with appended information,
such as the expert system mentioned above as an example, then encodes the
information using the most appropriate one of the suite of compression algorithms
available to him, and applies the appropriate compression method identifier to the
encoded information, to form a token. The token include the compressed information
to be transmitted, together with the decompression algorithm identifier. The token
is then appended to the mobile intelligent agent, and transmitted onto the network,
where it moves from place to place attempting to fulfill its preprogrammed
function.

Detailed Description Text (4):
As described above in relation to FIGS. 4 and 5, if the intelligent mobile agent
arriving at a network node is authorized to access the node, requires capability,
and the capability is available, the docking layer 312 of software provides access
to the capability by transferring a pointer to the intelligent mobile agent. In the
context of this invention, the capability which is made available to the
intelligent mobile agent may include a decompression algorithm identified by the
identifier accompanying the (or one of the) token(s) carried by the intelligent
mobile agent.

Detailed Description Text (5):
In FIG. 7, the dock software token processing starts at a START block 710, and
proceeds to a logic block illustrated as a block 712, which represents waiting for,
and recognizing the arrival of an intelligent mobile agent at the node. The logic
will remain in block 712 until an intelligent mobile agent arrives and is
recognized.

Detailed Description Text (6):
The arrival of an intelligent mobile agent at a dock is represented by block 810 of
FIG. 8. From logic block 810 of FIG. 8, the logic flows by way of a logic node A-1
to a logic block 712 of FIG. 7. When an agent arrives at the dock, the dock logic
of FIG. 7 flows from block 712 to a decision block 714, which represents the result
of a security test. If the agent has no authority to act in relation to the node,
the logic flows by the NO logic output of decision block 714 to a block 716, which
represents rejection of the agent, and the dock logic loops back to waiting block
712 by way of a logic path 718. If the intelligent mobile agent is authorized to
act, the logic leaves decision block 714 by the YES logic output, and flows to a
block 720, which represents a command to start the intelligent mobile agent.

Detailed Description Text (7):
From block 720 of FIG. 7, the logic flows by way of a logic node B-1 to block 812
of the agent logic flow of FIG. 8. Block 812 represents the starting of the docked
intelligent mobile agent. From block 812, the logic of the agent flows to a block
814, which represents transfer to the dock of a copy of each token carried by the
docked agent. From block 814, the logic flows or transfers by way of a logic node
A-2 to a block 722 of FIG. 7. Block 722 of FIG. 7 represents the reception of a
token at the dock. The logic flows from block 722 to a block 723, which represents
examination of the token to determine the type of algorithm by which the token was
coded. Block 724 represents the decoding or decompression of the token by the
appropriate one of the algorithms, to recover plain text. When the token has been
received and decoded, it is examined in a decision block 725 to determine if the
dock already has a copy of this token. In making the evaluation, the date on which
the token was generated should be taken into consideration; a different date may
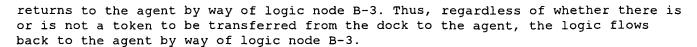represent a different token notwithstanding the same name.

Detailed Description Text (8):
If the token is not already available to the dock, the logic flows from the NO
output port of decision block 725 to a further logic block 726, which represents
the placing of the token into a memory for storage therein. The logic leaves block
726 and flows to the intelligent mobile agent by way of a logic node B-2. If the
token is already available to the dock when the decision is made in decision block
725 the logic flows to logic node B-2 by way of a block 728, which represents the
discarding of the superfluous token.

Detailed Description Text (9):
From logic node B-2 of the agent logic of FIG. 8, the logic flows to a block 816,
which represents a request by the agent directed to the dock for transfer to the
agent of appropriate tokens. In order to allow the dock to know what the agent
considers to be relevant, the request by the agent includes interest area
descriptors such as (a) age range of the token which is of interest to the agent,
(b) time or date of recording of the data in the token, (c) key-words, subjects or
terms of interest, or (d) token date or geographic range of interest. With such
information from the agent, the dock can readily select tokens containing the
desired information. From block 816, the logic is transferred by way of a logic
node A-3 to a block 730 of FIG. 7, representing examination at the dock of the
agent's request for tokens, to determine if any of the available tokens meet the
stated needs of the agent. From block 730, the logic flows to a decision block 732.
If a token is available which should be transferred to the mobile agent at its
request, the logic leaves decision block 732 by the YES output, and flows to a
further block 733, which represents the encoding or compression of the token to be
transferred to the mobile agent, if not already encoded. The logic then flows to a
block 736, which represents the transfer of the encoded relevant token to the
requesting agent. From block 736, the logic flows by way of a logic node B-3 to the
agent. If no token is available to the dock which should be transferred to the
intelligent mobile agent docked therewith, the logic of FIG. 7 leaves decision
block 732 by the NO output, and flows to a block 734, which represents the sending
of a NO RELEVANT TOKENS message to the docked agent. From block 734, the logic

returns to the agent by way of logic node B-3. Thus, regardless of whether there is or is not a token to be transferred from the dock to the agent, the logic flows back to the agent by way of logic node B-3.

Detailed Description Text (11):
The performing of the docked intelligent mobile agent's task is represented by block 822 of FIG. 8. Block 822 may be considered to be simply an interrupt, which is closed when the task has been completed. From block 822, the logic flows back to block 816, which, as mentioned above, represents a request to the dock for any additional relevant tokens, and continues on the logic path described following block 816. On the other hand, decision block 820 will ultimately decide that the intelligent mobile agent has no further tasks to accomplish at this location, and the logic flows from the NO output of decision block 824 to a block 824, which represents a command to the dock to send the intelligent mobile agent to the next location or network node.
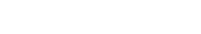
Detailed Description Text (12):
From block 824, the logic returns by a logic node A-4 to control by the dock, at block 738 of FIG. 7, which represents receipt of the request to move the intelligent mobile agent to a node, normally specified by the intelligent mobile agent. From block 738, the logic flows to a block 740, representing the sending of the requesting agent to the next network node. The logic of the dock then flows back to block 712.

Detailed Description Text (13):
Thus, a method according to the invention performs processing on a distributed network (10) of network nodes (14a, 14b, 14c, 14d) by, or with the use of, two or more intelligent mobile agents (310a, . . . , 310n). Each of the network nodes (14) at which processing may be accomplished includes an intermediary module (312) or dock program. The method includes various steps. Among the steps of the method is the step of selecting one compression algorithm from among the compression algorithms of a suite of compression algorithms, to thereby identify a selected compression algorithm. A compressed representation of a piece of information is generated by using the selected one of the compression algorithms. An information token is generated by attaching to the compressed piece of information a tag representing the identity of the selected compression algorithm used in the step of generating the compressed representation of the piece of information. The information token is attached to a first intelligent mobile agent. The first intelligent mobile agent is transmitted from a first one of the nodes (14a) in the system, together with its attached token, to at least a second node (14b, 14c, 14d, . . .) of the network (10). From a node in the network, which may be the first node or another node, a second intelligent mobile agent is transmitted to at least the second node of the network, whereby the first and second intelligent mobile agents reside at the second one of the nodes of the network, either simultaneously or sequentially. At the second node, the token is at least copied, or possibly transferred, from the first intelligent mobile agent to the intermediary module of the second node. In the intermediary module of the second node, the piece of information in the token is decompressed using the identity tag of the selected compression algorithm, to thereby produce a reconstituted piece of information. The reconstituted piece of information is copied or transferred to the second intelligent mobile agent. The second intelligent mobile agent then uses the reconstituted information to perform whatever task it is assigned to perform.

Detailed Description Text (14):
In a particular embodiment of the invention, the first mobile agent arrives at the second node at a time at which the second mobile agent is not yet at the node, and the method further includes the steps of, for (or at) the first mobile agent, resident at the second node, determining that the second mobile agent may eventually arrive at the second node, W and deciding to leave a copy of the token at the second node. In response to the decision to leave a copy of the token, the

first mobile agent leaves a copy of the token with the intermediary module at the second node. In a preferred embodiment of the invention, the intermediary module of the second node leaves the token compressed until that later time at which arrival, at the second node, is identified of an intelligent mobile agent which can make use of the piece of information. When arrival of such an intelligent mobile agent is identified, the piece of information is decompressed to produce the decompressed piece of information, using the identity tag of the selected compression algorithm to select the appropriate decompression algorithm. The reconstituted piece of information is then copied, or possibly transferred, to the second intelligent mobile agent.

Other Reference Publication (5):
"Telescript Technology: Mobile Agents", James E. White, General Magic, Inc., American Association for Artificial Intelligence, 1996 pp. 1-38.

CLAIMS:

1. A method for performing, by the use of plural intelligent mobile agents, processing on a distributed network of nodes, each of which includes an intermediary module, said method comprising the steps of:

selecting one compression algorithm from among the compression algorithms of a suite of compression algorithms to thereby identify a selected compression algorithm;

generating a compressed representation of a piece of information using said selected compression algorithm, to thereby generate a compressed piece of information;

generating an information token by attaching to said compressed piece of information the identity of said selected compression algorithm used in said step of generating a compressed representation;

attaching said information token to a first intelligent mobile agent;

from one of said nodes in the system, transmitting said first intelligent mobile agent, together with its attached token, to at least a second one of said nodes of said network;

from a node in said system, transmitting a second intelligent mobile agent to at least said second one of said nodes of said network, whereby said first and second intelligent mobile agents reside at said second one of said nodes of said network;
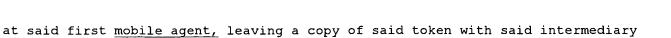
at said second node, at least copying said token from said first intelligent mobile agent to said intermediary module of said second node;

in said intermediary module of said second node, decompressing said piece of information in said token using said identity of said selected compression algorithm, to thereby produce a reconstituted piece of information;

at least copying said reconstituted piece of information to said second intelligent mobile agent.

2. A method according to claim 1, wherein said first mobile agent arrives at said second node at a time at which said second mobile agent is not at said node, said method further comprising the steps of:

at said first mobile agent, determining that said second mobile agent may eventually arrive at said second node, to thereby decide to leave a copy of said token at said second node;

at said first <u>mobile agent,</u> leaving a copy of said token with said intermediary module at said second node in response to said decision to leave a copy.

3. A method according to claim 2, wherein said method further comprises the steps of:

at said intermediary module of said second node, storing said token until that time at which arrival is identified of an intelligent <u>mobile agent</u> which can make use of said piece of information;

at said intermediary module, decompressing said piece of information in said token using said identity of said selected compression algorithm, to thereby produce at said time a reconstituted piece of information; and

at least copying said reconstituted piece of information to said second intelligent <u>mobile agent</u>.

First Hit   Fwd Refs

☐ ░░░ Generate Collection ░░░

L9: Entry 2 of 8   File: USPT   Mar 9, 2004

DOCUMENT-IDENTIFIER: US 6704765 B1
TITLE: System for allocating resources among agent processes

Abstract Text (1):
A certain number of agents and tokens are created. A first agent that is assigned a
request must obtain a token to process the request. The number of agents which may
concurrently process requests are limited by preventing an agent from processing a
request unless the agent has an assigned token. An additional agent is created
after assigning the token to the first agent, provided that an existing agent is
not available to wait for a request, and an unassigned token is available for the
additional agent. A certain minimum number of agents is defined and an agent is
automatically eliminated when the agent completes a request, provided that the
number of agents that are assigned tokens exceeds the minimum number of agents.

Application Filing Date (1):
19941214

Parent Case Text (2):
This application is related to co-pending application entitled (1) "System for
Coordinating the Assignment of Agent Processes to Data Processing Requests," Ser.
No. 08/355,861, filed Dec. 14, 1994 and Ser. No. 08/376,707, filed on Jan. 31, 1995
respectively assigned to the same assignee.

Brief Summary Text (9):
According to the present invention, the foregoing and other objects are attained by
creating a certain number of agents and tokens. A first agent that is assigned a
request must obtain a token to process the request. The number of agents which may
concurrently process requests is limited by preventing an agent from processing a
request unless the agent has an assigned token.

Drawing Description Text (10):
FIG. 9 shows categories of agents and shows counters used in coordinating the
agents.

Detailed Description Text (19):
During run-time, a user application program 26A1 generates an application request
100A1 corresponding to one of the application requests 100A1 that has been compiled
into low level requests 108A1. The DBM program 102A has a database manager queue
("DBM queue") 114A for storing application requests 100A and a parallel database
queue ("PDB queue") 116A for internal requests 118A generated from the low level
requests 108A. Request 100A1 is put into the DBM queue 114A along with the name of
user that originated the request. An agent 120A assigned to DBM queue 114A from the
pool 104A of agents 104A1, 104A2, etc. is assigned to a request 100A1. The low
level requests 108A1 compiled from application request 100A1 are stored in a
portion of the access plan 110A1 for the application program 26A1 which generated
the request 100A1. The coordinating agent 120A assigned to the request 100A1 gets
the low level requests 108A1 and combines them with addressing information from the
system configuration and access authorization information derived from the system
configuration and from the user name, thereby generating an internal request 118A
for handling by an agent 122A assigned to PDB queue 116A from the pool 104A of

agents 104A1, 104A2, etc.

Detailed Description Text (20):
All of the agents 104A1, 104A2, etc. in the pool 104A are capable of servicing
either the application requests 100A from the DBM queue 114A or the internal
requests 118A from the PDB queue 116A. While an agent is servicing a request from
the DBM queue it is referred to as a "coordinating agent". While an agent is
servicing a request from the PDB queue, the agent is referred to as a "subordinate
agent".

Detailed Description Text (21):
Since the coordinating agent 120A provides addressing information and access
authorization to the compiled requests 108A, the application program and the
compiler may be independent of the system configuration, and a subordinate agent
may process interval requests independently of the application users. Since the DBM
program 102A, coordinate agent 120A, and subordinate agent 122A exchange only those
requests which have been compiled from external application requests 100A1, 100A2,
etc., and have been supplemented with access authorization, interagent
communication does not require detailed error checking, such as that which a
transport layer communications protocol ordinarily provides. The DBM program 102A
therefore provides a fast communications process (referred to herein as the "fast
communications manager" or "FCM") 124A with reduced error checking for coordinating
communication between agents.

Detailed Description Text (27):
Returning now to the description of agent message processing, after initialization,
as shown in FIG. 4, requests 100A and 118A are serviced by the agents 104A.
Messages are identified and stored for passing among agents according to the
following system of queues and addressing (as illustrated in FIG. 7). An AMQ 130A1
is designated for a particular request 118A1 and related subsequent requests by
storing the request's UUID 112A1 in the AMQCB 132A1 of the AMQ 130A1 so that the
AMQ 130A1 may be located by reference to the UUID 112A1. That is, for example,
after a coordinating agent 120A generates an internal request 118A, the FCM 124A
processes the internal request 118A1, which includes the FCM putting the internal
request 118A into an AMQ 130A1, putting the request's UUID 112A1 into the AMQ's
AMQCB 132A1, and putting the communication handle 150A1 for the AMQCB 132A1 into
the PDB queue 116A. Referring now to FIG. 6, in selecting an AMQ 130A1 for an
internal request 118A1, the FCM 124A searches among the AMQCB's 132A4 and 132A5 of
the busy AMQ's 130A4, and 130A5 attempting to find the UUID 112A1, of the internal
request 118A1. If the FCM 124A finds the UUID 112A1 in an AMQCB 132A4 or 132A5, the
FCM 124A puts the request 118A1 into the associated AMQ 130A4 or 130A5. Otherwise,
the FCM 124A puts the request 118A1 into a free AMQ 130A1, 130A2 or 130A3.

Detailed Description Text (31):
A token scheme limits the number of agents which may process requests concurrently.
That is, a network administrator sets the total number of tokens available and an
agent must get a token in order to process a request. Variations on this scheme are
contemplated wherein different tokens have different priority values so that
throughput or response time may be optimized based on historical input/output
activity and processor load. An agent may return its token either upon finishing
processing or when the agent suspends processing temporarily, such as while waiting
for a resource.

Detailed Description Text (33):
A network administrator sets an initial number of agents in 104A1, 104A2 . . .
104B1, 104B2 . . . etc. the pools of agents 104A, 104B, etc. for each node and sets
a target minimum number of agents in each pool. Thereafter, the number of agents in
the pools may be automatically increased or decreased by the DBM program 102. When
all the agents 104 are processing requests 118, and an additional request is
received, a new agent 104 is automatically created, provided that enough tokens are

available. If, when an agent 104 finishes processing a request 118, there are more agents 104 than the initial number, and if an analysis of the history of requests provides a certain indication, then an agent 104 will terminate itself. The analysis of the history of requests functions to dampen the termination of agents 104. It has a particularly significant effect to terminate an agent 104 when the number of agents 104 is near the minimum number.

Detailed Description Text (35):
Referring now to FIG. 9, there are four categories of agents 104. First, there is the category of agents which are actively processing a request, or "active" agents 170. In this example, for node 10A there are nine agents 104A1 through 104A9, and attached agents in the other three categories are organized in queues. One queue, the free agent queue 172, is of "free" agents which are waiting for a request. In this example, agents 104A7 through 104A9 are shown as agents in the free agent queue 172A. Another queue, the resume agent queue 174, is of "resuming" agents. In this example, agents 104A3 and 104A5 are shown as agents in the resume agent queue 174A. An agent in the resume agent queue 174A has a request, is not waiting for a resource or event, but is waiting for a token to allow it to begin processing the request. Finally there is the category of blocked agents 176. In this example, agent 104A4 is shown as an agent in the blocked agent queue 176A. An agent in the blocked agent 176A queue has a request but is waiting on a dependency to be resolved, that is, it is waiting on a resource or event. Each agent has an Agent Control Block ("ACB") (not shown) with a flag (not shown) indicating whether an agent is in an active, wait, or sleep state. The free agent queue 172 and resume agent queue 174 are implemented in the DBM program. The blocked agent queue 176 is implemented at the operating system level.

Detailed Description Text (52):
Upon getting a token, the agent begins processing the request. If the agent suspends processing of the request, such as while waiting on a resource or event, then after a predetermined period of time the agent will give up its token and attempt to notify an agent on the resume agent queue. Also, when the agent completes processing of the request, the agent will give up its token and attempt to notify an agent on the resume agent queue. These operations where a token is released are described further in the following pseudo code:

CLAIMS:

1. A method for allocating computer processing resources on a node among a number of concurrent processes ("agents") which share the resources for processing a number of requests, comprising the steps of: creating a certain number of agents on the node; assigning a first request to a first agent for processing of the request; creating a certain number of tokens; assigning a token to each agent having an assigned request; and limiting the number of agents which may concurrently process request by preventing an agent from processing a request unless the agent has an assigned token.

7. A method for allocating computer processing resources on a node among a number of concurrent processes ("agents") which share the resources for processing a number of requests, comprising the steps of: creating a certain number of agents; assigning a request to an agent for processing of the request; creating a certain number of tokens; assigning a token to a first agent having an assigned request; limiting the number of agents which may concurrently process requests by preventing an agent from processing a request unless the agent has an assigned token; enqueuing a second agent in a resume agent queue when the second agent is assigned a request and a token is not available so that the second agent may await notification of an available token; releasing the assigned token by the first agent when the first agent completes the assigned request so that the token may be reassigned to an agent having a request to process; notifying the second agent by the first agent, if the first agent releases the token; creating an additional

agent after assigning the token to an agent, if another agent is not available to wait for a request, and an unassigned token is available for the additional agent; defining a certain minimum number of agents; and automatically eliminating an agent when the agent completes a request if the number of agents that are assigned tokens exceeds the minimum number of agents.

8. A computer system including a plurality of processors for allocating system resources among a number of agent Processes which share the system resources for processing a number of requests, comprising: a plurality of agent,processes in a system memory; means for assigning a request to a first agent process for processing of the request; a plurality of tokens in the system memory; means for assigning a token to the first agent process; and means for limiting the number of agent processes which may concurrently process requests by preventing an agent process from processing a request unless the agent process has an assigned token.

□ ▨▨▨ Generate Collection ▨▨▨

L9: Entry 4 of 8   File: USPT   Sep 11, 2001

DOCUMENT-IDENTIFIER: US 6289394 B1
TITLE: Agent management system capable of readily monitoring and controlling agent

Application Filing Date (1):
19950217

Brief Summary Text (25):
Moreover, it is not possible either to join any existing application not being made
to operate in an autonomous coordinating operation system to the coordinating
operation likewise other agents, or to monitor and control such applications in the
same manner.

Brief Summary Text (32):
In view of the above object, an agent management system according to this invention
comprises an agent for executing autonomous coordinating processes, first
information interchanging means for interchanging information necessary for the
autonomous coordinating processing of the agent to/from the agent, agent-
monitor/control request means for requesting control and monitoring of the agent
from outside, and second information interchanging means for interchanging
information required by the agent-monitor/control request means to/from the agent
by synchronous communication. Accordingly, it is possible to control the internal
information of the agent through the second information interchanging means,
besides the conventional autonomous coordinating operation to be carried out by the
agent independently.

Brief Summary Text (36):
According to a fifth aspect of this invention, the agent management system further
comprises monitoring/controlling method executing means for executing methods for
monitoring/controlling the agent at the processing level in reply to a request from
the agent-monitor/control request means, monitoring method means for storing a
monitoring method to be common in a plurality of agents and for being executed by
the monitoring/controlling method executing means, and controlling method means for
storing a controlling method to be common in a plurality of agents and for being
executed by the monitoring/controlling method executing means. As a result, the
agents can be targets of monitoring and controlling operations even if each agent
has no data or method.

Brief Summary Text (37):
According to a sixth aspect of this invention, an agent management system comprises
an application not prepared as an agent operating in an autonomous coordinating
environment, information-reading interface means for reading information on behalf
of the application, information-writing interface means for writing information on
behalf of the application, first storage means for storing formats of information
dependent on the application, second storage means for storing formats of
information dependent on the communication of an autonomous coordinating operation
system, information format converting means for converting the information format
of the communication used in the application into an information format of
communication used in the autonomous coordinating operation system on the basis of
the information format stored in the first and the second storage means, said
converting means being connected to the application through the information-reading

and the information-writing interface means, an information interchanging field for interchanging information in the autonomous coordinating environment, transmitting means for transmitting information received from the application to the information interchanging field through the information format converting means, and receiving means for transmitting information received from the information interchanging field to the application through the information format converting means. Therefore, even an existing application not having been made to operate in the autonomous coordinating environment can join the autonomous coordinating operation environment as a pseudo-agent.

Brief Summary Text (38):
According to a seventh aspect of this invention, an agent management system comprises an application not having been prepared as an agent operating in an autonomous coordinating environment and having standard input/output ports, first storage means for storing an information format dependent on the application, second storage means for storing an information format dependent on communication in the autonomous coordinating environment, information format converting means for converting the information format of the communication used in the application into an information format used in the autonomous coordinating operation system on the basis of the information format stored in the first and the second storing means, said converting means being connected to the application through the information-reading interface means and the information-writing interface means, an information interchanging field for interchanging the information in the autonomous coordinating environment, transmitting means for transmitting information received from the application to the information interchanging field through the information format converting means, and receiving means for transmitting the information received from the information interchanging field to the application through the information format converting means. As a result, if existing applications just have standard input/output ports, they can readily join the autonomous coordinating operation environment as pseudo-agents.

Brief Summary Text (41):
According to a ninth aspect of this invention, an agent management system further comprises event generating means for transmitting pseudo-events to the transmitting route being monitored by the event monitoring means. Accordingly, the existing applications can readily join the autonomous coordinating environment as agents without the need to be modified.

Brief Summary Text (44):
According to a twelfth aspect of this invention, an agent management system further comprises a plurality of autonomous coordinating operation environments each of which has an agent, policy managing means for managing policies of the plurality of autonomous coordinating operation environments, connected state managing means for managing the connected states of the plurality of autonomous coordinating operation environments, and inter-environments information interchanging means for interchanging the information among the autonomous coordinating operation environments in accordance with the policy managing means and the connected state managing means. As a result, it is readily possible to constitute a system by combining a plurality of distributed autonomous coordinating operation environments.

Detailed Description Text (3):
FIG. 5 is a block diagram showing an agent management system according to a first embodiment of this invention. In FIG. 5, the numerals designate respectively: 19, agent-monitor/control request means; 18, an autonomous coordinating operation environment including the agent 1 and an information interchanging field 15; 2, a computing section of the agent 1; 3, an input/output processing section for processing input/output information in the agent 1; 7, internal information acquisition means for acquiring internal information in the computing section 2; 4, internal information including internal data 5 and internal method 6 in the

computing section 2; 8, internal information setting means for setting the internal information 4 in the computing section 2; 14, an information output section of the input/output processing section 3; 13, an information input section of the input/output processing section 3; 17, second information interchanging means, as a significant feature of this invention, for interchanging information for monitoring/controlling operations through synchronous communication in the information interchanging field 15; 16, first information interchanging means for interchanging information necessary for the autonomous <u>coordinating processing</u> <u>to/from the agent</u> through asynchronous communication; 12, second information interchanging means for interchanging information required by the agent-monitor/control request means to/from the agent through the synchronous communication; 11, autonomous distributed-system asynchronous communication output means; 9, autonomous distributed-system asynchronous communication input means for inputting information of autonomous distributed system through asynchronous communication; and 10, controlling/operating synchronous communication input means for inputting controlling and operating information through the synchronous communication. The agent-monitor/control request means 19 is connected to the second information interchanging means 17. The second information interchanging means 17 is also connected to the monitoring/operating synchronous communication output means 12 and the controlling/operating synchronous communication input means 10. The first information interchanging means 16 is connected to the autonomous distributed-system asynchronous communication output means 11 and to the autonomous distributed-system asynchronous communication input means 9. The monitoring/operating synchronous communication output means 12 is also connected to the internal information getting means 7. The autonomous distributed-asynchronous communication output means 11 is also connected to the internal data 5 of the internal information 4. The autonomous distributed-system asynchronous communication input means 9 is also connected to an internal method 6 of the internal information 4. The controlling/operating synchronous communication input means 10 is also connected to the internal information setting means 8. The internal information getting means 7 and the internal information setting means 8 are connected to the internal information 4 in the computing section 2. There are a plurality of agents 1 in the autonomous coordinating operation environment 18.

<u>Detailed Description Text</u> (45):
According to the above-mentioned structure, the existing application 34 can take part in the coordinating operation in the autonomous <u>coordinating operation system</u> <u>as a pseudo-agent</u> 1 just by exchanging the input/output processing section 3 with the computing section 2 left intact.

<u>Detailed Description Text</u> (85):
Furthermore, it is also possible to make the existing application not being composed as an agent operating in the autonomous coordinating environment act as a similar <u>coordinating operation to the other autonomous agents,</u> such that existing applications can be reused in the autonomous coordinating operation environment, so as to further improve the efficiency of the system construction.
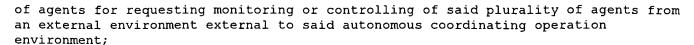
CLAIMS:

1. An agent management system comprising:

a plurality of agents, each for performing an autonomous coordinating process within an autonomous coordinating operation environment;

first information interchanging means associated with each said agent within said autonomous coordinating operation environment for interchanging information necessary for the autonomous <u>coordinating process of said agent</u> by asynchronous communication to/from said agent;

agent-monitor/control request means external to and associated with said plurality

of agents for requesting monitoring or controlling of said plurality of agents from an external environment external to said autonomous coordinating operation environment;

second information interchanging means associated with each said agent within said autonomous coordinating operation environment for interchanging necessary information requested by said agent monitor/control request means by synchronous communication to/from said plurality of agents within said agent management system; and

said first information interchanging means suspending the asynchronous communication to/from said agents while said second information interchanging means synchronously communicates to/from said agents so that said agent-monitor/control request means can directly monitor and control said agents from the external environment.

7. An agent management system comprising:

a plurality of applications not being composed as an agent each of which operates in an autonomous coordinating environment;

standard input/output and error output ports associated with each application;

first storage means associated with each application for storing an information format dependent on said application;

second storage means associated with each application for storing an information format dependent on the communication in the autonomous coordinating operating system;

information format converting means associated with each application for converting, the information format of communication used in said application into an information format of communication used in the autonomous coordinating operation system, based on the information stored in said first and second storage means, said information format converting means being connected to each application through an information-reading interface means and an information-writing interface means;

an information interchanging field external to and associated with said plurality of applications where the information is mutually interchanged in the autonomous coordinating environment;

transmitting means associated with each application for transmitting the information obtained from said application through said information format converting means to said information interchanging field;

receiving means associated with each application for receiving the information from said information interchanging field to said application through said information format converting means; and

joining means for joining existing applications not composed as agents which operate in an autonomous <u>coordinating environment as a plurality of pseudo-agents,</u> wherein the existing applications only include standard input/output and error output parts.

12. An agent management system comprising:

a plurality of autonomous <u>coordinating operation environments, each of which</u> <u>includes an agent</u>;

a policy managing means external to and associated with said plurality of
autonomous coordinating operation environments for managing policies of said
plurality of autonomous coordinating operation environments;

a connecting state managing means external to and associated with said plurality of
autonomous coordinating operation environments for managing connecting states of
said plurality of autonomous coordinating operation environments at discrete time
intervals; and

an inter-environmental information interchanging means external to and associated
with said plurality of autonomous coordinating operation environments for
interchanging information among said plurality of autonomous coordinating operation
environments in accordance with said policy managing means and said connecting sate
managing means; wherein said inter-environmental information interchanging means
will not transfer information to an autonomous coordinating operation environment,
when the information violates a policy of the autonomous coordinating environment.

First Hit

□ Generate Collection

L6: Entry 3 of 3   File: DWPI   May 24, 2004

DERWENT-ACC-NO: 1998-586363
DERWENT-WEEK: 200434
COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Mobile agent and message to computer transmission method – use temporary computer for supplementary storage area when direct addressing to destination is not permitted

Basic Abstract Text (2):
If it is determined that the second computer cannot receive the sequence i.e when the mobile agent cannot move directly to the destination computer, an address of the computer and the bit sequence is transmitted to the temporary computer. At an appropriate time the destination computer extracts the agent and the message addressed to it from the temporary storage [21].

Basic Abstract Text (3):
ADVANTAGE - Permits a mobile agent to move to a destination computer regardless of the condition of the destination computer.

PF Application Date (1):
19970530

PF Application Date (3):
19980327

PF Application Date (4):
19970530

PF Application Date (5):
19980414

PF Application Date (7):
19980327

PF Application Date (8):
19980414

Equivalent Abstract Text (2):
If it is determined that the second computer cannot receive the sequence i.e when the mobile agent cannot move directly to the destination computer, an address of the computer and the bit sequence is transmitted to the temporary computer. At an appropriate time the destination computer extracts the agent and the message addressed to it from the temporary storage [21].

Equivalent Abstract Text (3):
ADVANTAGE - Permits a mobile agent to move to a destination computer regardless of the condition of the destination computer.

Standard Title Terms (1):
MOBILE AGENT MESSAGE COMPUTER TRANSMISSION METHOD TEMPORARY COMPUTER SUPPLEMENTARY STORAGE AREA DIRECT ADDRESS DESTINATION PERMIT

as the management console 110. Thus, RMON enables a management computer, such as the management console 110, to monitor a network as a whole rather than just individual network devices on the network. As shown in FIG. 2D, an RMON software module 230 is loaded into the memory 124 of the network device 102 and activated by the MADC 220. The MADC 220 or the new RMON software module 230 includes the following BRACH structure RMONBranch, which describes a table to be added to the MIB 206 for holding the RMON objects:

Detailed Description Text (23):
In this manner, after new objects are dynamically added to the MIB 206, a resource manager or management station, such as the management console 110 of the network system 100, sends SNMP requests, such as get, getnext or set, to the management module 116 of the network device 102 managing a new device, such as the SUM 132, where the requests are received by an agent, such as the MADC 200 or 220 or other software module, such as one of the upgrade modules 208, a software module 222, the RMON module 230, etc. The request or requests include OIDs to one or more objects to be read or modified as desired. The agent interfaces with the MIB manager 202, which executes the corresponding functions in response to the SNMP requests, such as query, modify, etc. to retrieve or modify the information as desired. For a set operation, the MIB manager 202 retrieves the object values identified by OID from the MIB and provides the information to the agent, which reports back to the management console. For a get operation or getnext operation, the MIB manager 202 modifies the object values as requested. If values are monitored according to a trap operation, the MIB manager 202 provides updated object values from the MIB to the agent, which provides the updated values to the management console.

CLAIMS:

2. The MIB manager of claim 1, wherein said plurality of functions includes an add function called with an address to a location within the agent to a definition of a new object for adding the new object to the MIB structure.

22. The network system of claim 21, further comprising:

one of said plurality of network nodes including a management console for submitting requests to said agent of said network device.

☐ ▓▓▓ Generate Collection ▓▓▓ | Print |

L70: Entry 6 of 21   File: USPT   Jun 15, 1999

DOCUMENT-IDENTIFIER: US 5913037 A
TITLE: Dynamic management information base manager

Application Filing Date (1):
19960703

Brief Summary Text (7):
In general, some sort of network or resource manager, such as a software agent,
implements an MIB. For example, a software agent operating on a network switch or
repeater maintains an MIB for managing that network device. A management console
operating on a computer system in the network may monitor and manage a network
device by sending SNMP requests to a software agent running on the device, where
the agent accesses its local MIB to retrieve or modify MIB objects. The SNMP
request includes one or more OIDs to the objects in the MIB of interest. For
example, the management console sends a "get", "getnext" or "set" operation to a
software agent across the network or other communications path, which accesses the
MIB and responds by reading or modifying, respectively, one or more objects
identified by the OIDs in the MIB according to the specific operation. The get
operation is used to read the value of an object identified by an OID and the
getnext operation is used to read the value of the next object or "leaf" in the MIB
tree referenced by a given OID. The set operation is used to modify the value of an
object identified by an OID. A "trap" operation is similar to an interrupt, where
if the value of an object changes, the software agent respondingly sends the new
value to a management console.

Brief Summary Text (17):
The management module describe above may be used in any network or non-network
device, where it is desired to manage certain aspects and parameters of that
device. A network device, such as a repeater or switch, may further include
hardware options for modifying or upgrading the network device. For example, a fast
Ethernet repeater may include a slot for receiving and coupling an uplink device
for expanding the network. The software agent managing the device may include
object definitions, procedures and functions for managing the uplink device. In a
more modular approach, a new software module is added to upgrade the agent, where
the new software module includes the object definitions and the management code. In
either case, the agent or the new module calls the functions within a MIB manager
according to the present invention, which updates the MIB structure accordingly to
enable management of the uplink device. A management station or node in the network
may further include a management console for submitting SNMP requests to the agent.
In this manner, the management console queries the status of the network device,
and, if installed, the status of the uplink device.

Detailed Description Text (19):
FIG. 2D illustrates another example for adding elements to the MIB 206 provided by
a software subsystem, which, in this example, is the remote network monitoring
(RMON) subsystem. A standard MIB, such as MIB-II, enables management of a
particular device, node or single point of attachment in a network. An RMON
subsystem expands the management capabilities and provides an effective and
efficient way to monitor network-wide behavior of multiple nodes and devices, while
reducing the burden both on other software agents and on management stations, such

First Hit   Fwd Refs

☐ ▓▓▓ Generate Collection ▓▓▓

L51: Entry 1 of 15   File: USPT   Feb 29, 2000

DOCUMENT-IDENTIFIER: US 6032142 A
TITLE: Autonomous compound agents

Application Filing Date (1):
19971105

Detailed Description Text (3):
FIG. 1 shows an interactive entertainment apparatus based around a rule-based data
processing apparatus embodying the present invention. The apparatus comprises a
processor 10 operating on a compiled asynchronous logic circuit representation
defined by a population of RTA rules. Through control and data link 12, the
processor 10 is enabled to control a multi-character interactive animation for
which scenes are generated by image rendering stage 14 and output to display 16.
The processor 10 has associated with it a number of data stores, a first of which
is random access memory (RAM) 18 which provides discrete storage areas for a table
20 of the various elements of the asynchronous logic circuit representation of the
compiled population of RTA rules, and listings of propagate functions 22, future
element state changes 24, and an event stack 26. These features perform their
conventional functions, a more detailed description of which is given in the above-
mentioned European patent application EP-A-0,704,077.

Detailed Description Text (4):
The RAM 18 is connected to the processor 10 by a data bus 28 and an address bus 30
in known manner. Also in known manner, a clock (CLK) 32 is connected to the
processor and, as shown, the rendering stage 14. A read-only memory (ROM) 34 is
coupled to the processor 10 by a data bus 36 and address bus 38 and provides a
program store holding a command structure governing implementation of the compiled
circuit representation for each of the compound and component agents. A user
interface device (UID) 40 is provided to enable input of control data and commands
directly to the processor 10; the form of the UID will depend on what is required
by the application, and may range from one or two simple buttons to keyboard/mouse
input in conjunction with a graphical user interface on display 16 and generated by
the rendering stage 14.

Detailed Description Text (23):
As well as interacting with other component agents within the compound agent,
component agents can interact (via markers) with other objects and compound agents
in their environment. Thus each component agent may in itself display skilful
behaviour with respect to other objects or compound agents. The markers on the
objects of interest to the skill agents correspond to the parameters of the
behaviours or the objects towards which the behaviours are directed (such as the
door, in the case of the "going to" and "turning towards" component agents). The
top decision structure of each component agent provides a way of deciding which
object the activity of that component agent is directed towards when there are
competing requests from other component agents for assistance.